# Residuals-Based Subgraph Detection with Cue Vertices

*(Invited Paper)*

Benjamin A. Miller, Stephen Kelley, Rajmonda S. Caceres, and Steven T. Smith
Lincoln Laboratory
Massachusetts Institute of Technology
Lexington, Massachusetts 02420
Email: {bamiller, stephen.kelley, rajmonda.caceres, stsmith}@ll.mit.edu

***Abstract*—A common problem in modern graph analysis is the detection of communities, an example of which is the detection of a single anomalously dense subgraph. Recent results have demonstrated a fundamental limit for this problem when using spectral analysis of modularity. In this paper, we demonstrate the implication of these results on community detection when a cue vertex is provided, indicating one of the vertices in the community of interest. Several recent algorithms for local community detection are applied in this context, and we compare their empirical performance to that of the simple method used to derive the theoretical spectral limits.**

## I. Introduction

In many applications, the data of interest take the form of entities and the relationships between them. These may represent a broad, diverse set of data types, from communication between people to interactions between proteins. In all of these diverse contexts, the relational data are typically represented as a graph.

One of the common problems among analysts working with graph-based data is subgraph detection. Given a large set of entities and their relationships, connections, and interactions, it can be difficult to determine if there is a particular subset of entities that requires special attention [1], [2]. Typically, the objective is to find a relatively small set of vertices whose topology is inconsistent with some notion of expected behavior in the graph. The classical planted clique problem embodies this in a simple form.

In the planted clique problem, the objective is to locate a subset where all possible connections exist, when connections across the rest of the graph occur with a fixed probability. This simplified scenario has enabled the derivation of hard detectability limits [3], [4]. While simplified for mathematical tractability, this problem yields valuable insight into detectability in more complicated networks derived from real data.

The planted clique problem is traditionally focused on uncued detection, i.e., determining the nodes that comprise the clique without any additional information about which entities are interesting. In practice, however, typically some additional knowledge priors are available. For example, in an advertising application in a social network, a company may have knowledge that a person uses their product, and wants to advertise to other network users who have close relationships with their current customer. Upfront knowledge priors enable more efficient use of resources by targeting a search that could consider the entire graph by having it prioritize entities in the graph that are close to the cue. Understanding the implications that recent subgraph detection bounds have on the setting where a cue is present will improve our understanding of detectability in this common alternative context.

In this paper, we investigate the implication of recent spectral limits of planted clique detection to cases where one entity in the clique is revealed. Using a simple method to reduce the number of entities considered, we can directly apply current bounds for uncued detection to the reduced dataset. The resulting bounds show that, under the right circumstances, it is possible to detect a clique that reduces its size as the overall graph gets larger. We demonstrate empirically that current cued subgraph detection methods go through a detectability phase transition at the same point as the simple filteringmethod, suggesting that the analysis applied here has implications for performance using several different methods.

The remainder of this paper is organized as follows. Section II formalizes the problem and defines our notation. In Section III, we review the recent spectral bounds on uncued planted clique detection, and their extension to planted dense subgraph detection. Section IV derives an extension of these results to cases where a cue provides a simple up-front entity filtering method. In Section V, we define a set of experiments in which we compare several cued subgraph detection algorithms from the open literature to the simple filtering method, and Section VI outlines the results of these experiments. We conclude the paper in Section VII with a brief summary and directions for future work.

## II. Problem Model

### A. Definitions and Notation

In the problem we consider, we are given a graph $G = (V, E)$, which is comprised of a set of vertices $V$ (representing entities), and a set of edges $E$ (the relationships between the entities). We denote the number of vertices in the graph by $N = |V|$. There is inherently a subgraph of interest, whose

This work is sponsored by the Assistant Secretary of Defense for Research & Engineering under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

vertices are denoted by $V_S \subset V$, and its size is denoted $k = |V_S|$. The graphs considered in this paper are unweighted (meaning connections either exist or do not, with no notion of magnitude) and undirected (meaning all connections are bidirectional).

Since the bounds we derive are based on spectral methods, we will make use of matrix representations of the graph. The adjacency matrix $A = \{a_{ij}\}$ of the graph $G$ is an $N \times N$ matrix where $a_{ij}$ is nonzero only if there is an edge in $E$ between vertices $v_i$ and $v_j$. (This requires an arbitrary labeling of the vertices with integers from 1 to $N$.) Since $G$ is unweighted, $A$ will be binary, and since $G$ is undirected, $A$ will be symmetric. The degree of $v_i$ (the number of edges connected to it) is denoted by $d_i$.

Other matrix representations of graphs have been used in the community detection literature. The graph Laplacian has been used to approximate the solution to the min-cut problem, where the objective is to make the graph disconnected by removing the smallest number of edges. The Laplacian is defined as

$$L := D - A, \tag{1}$$

where $D$ is a diagonal matrix where the entry in row $i$ and column $i$ is $d_i$. When there is some notion of the probability of connections, the modularity matrix has also been used for community detection [5]. This matrix is used to optimize the partition of a graph according to a different criterion: creating a partition where there are a greater-than-expected number of edges on either side of the partition, and fewer edges than expected crossing it. The modularity matrix is defined as

$$B := A - \mathbb{E}[A]. \tag{2}$$

Thus, $B$ represents the residuals obtained when subtracting the expected adjacency matrix from the observed. In the traditional planted clique problem, the background graph is an Erdős-Rényi random graph, i.e., a graph where each pair of vertices shares an edge with equal probability $p$.

### B. Cued Subgraph Detection

In the cued subgraph detection problem, we observe the graph $G$ and are given a cue vertex $v_c \in V_S$. Our objective is to determine the remainder of $V_S$. This is typically done by computing a test statistic $z(v)$ for each $v \in V \setminus \{v_c\}$, and estimating the subgraph of interest to be

$$\hat{V}_S = \{v_c\} \cup \{v \in V \setminus \{v_c\} : z(v) > t\},$$

where $t$ is a threshold that can be varied. Each of the algorithms we consider in Section V follows this format. In this paper, we evaluate performance based on receiver operating characteristic (ROC) metrics. Here, empirical probability of detection is

$$p_d = \frac{|V_S \cap \hat{V}_S| - 1}{|V_S| - 1}$$

(where 1 is subtracted in the numerator and denominator since we do not account for the cue vertex in the evaluation), the empirical false alarm rate is

$$p_{fa} = \frac{|\hat{V}_S \setminus V_S|}{|V \setminus V_S|},$$

and overall performance of a detection algorithm is evaluated based on the area under the ROC curve (AUC).

## III. Uncued Subgraph Detection Bounds

We will specifically consider recent spectral bounds for planted clique detection [4]. This work proposed a simple algorithm for planted clique detection by thresholding the principal eigenvector of the modularity matrix $B$, and showed that there is a sharp detectability threshold that can be derived via a random matrix theoretic analysis of the problem. The algorithm is as follows. Compute the principal eigenvector $u$ of the modularity matrix $B$, computed with respect to an Erdős-Rényi random graph. The estimated subgraph is then computed as

$$\hat{V}_S = \left\{ v_i : |\sqrt{N} u_i| > F^{-1}_{\mathcal{N}(0,1)} \left(1 - \frac{\alpha}{2}\right) \right\}, \tag{3}$$

where $F^{-1}_{\mathcal{N}(0,1)}$ is the inverse cumulative density function of a standard normal distribution and $\alpha$ is the desired false alarm rate. This algorithm is based on the fact that the modularity matrix of a planted clique in an Erdős-Rényi graph is well approximated by a rank-1 perturbation of a Wigner matrix (a symmetric random matrix where all entries have zero mean and equal variance), which has a known eigenvalue distribution that enables an analytical detection bound. The entries in the eigenvectors of a Wigner matrix also appear normally distributed as $N \longrightarrow \infty$. These observations yielded the following theorem.

*Theorem 3.1 (Nadakuditi [4]):* Consider a $k$-vertex clique planted in an $N$-vertex graph with edge probability $p$, where the clique vertices are identified using (3) for a significance level $\alpha$. Then, for fixed $p$, as $k, n \to \infty$ such that $k/\sqrt{n} \to \beta \in (0, \infty)$ we have

$$\mathbb{P}(\text{clique discovered}) \xrightarrow{a.s.} \begin{cases} 1 & \text{if } \beta > \beta_{\text{crit.}} := \sqrt{\frac{p}{1-p}} \\ \alpha & \text{otherwise.} \end{cases} \tag{4}$$

The detection threshold is based on the relationship between the nonzero eigenvalue of the rank-1 perturbation, $\theta = k(1-p)$, and the maximum eigenvalue of the random background, $R = \sqrt{4Np(1-p)}$. This can easily be extended to cases where a dense subgraph is embedded rather than a clique. If the subgraph has a probability of internal connection $p_{in} > p$, it will be detectable if

$$k(p_{in} - p) > \sqrt{Np(1-p)} \tag{5}$$

as $N \longrightarrow \infty$. Note that, if $p \longrightarrow 0$ as $N \longrightarrow \infty$, then the right hand side of (5) will approach the square root of the average degree. The left hand side will approach the average internal degree of the subgraph if $p = o(p_{in})$, and will approach a constant multiple of this quantity if $p = \Theta(p_{in})$.

## IV. Cued Subgraph Detection Bounds

### A. Cued Planted Clique Detection Setting

We will start by considering the planted clique problem when one of the clique vertices is revealed. Since all possible edges exist between clique vertices, we know that all clique vertices are in the one-hop neighborhood of the cue vertex. We denote by $N_i(v)$ the $i$-hop neighborhood of $v$, i.e., the vertices that can be reached from $v$ via a path of length $i$ or less. This allows a simple filtering procedure to incorporate the cue: We can consider only $N_1(v_c)$ rather than all of $V$. Since the edges in the background are all independent, when considering the induced subgraph of $N_1(v_c)\backslash\{v_c\}$ (i.e., the graph consisting of all edges in $E$ that occur between the vertices in the subset), the objective it to solve another planted clique problem. In this case, the clique has $k-1$ vertices, and the size of the background follows

$$|N_1(v_c) \setminus \{v_c\}| = k - 1 + \hat{d}_c, \tag{6}$$

where $\hat{d}_c$ is drawn from the binomial distribution $\mathcal{B}(N-k,p)$. We can use this fact to derive bounds for the cued case when applying the simple spectral algorithm to the cue's one-hop neighborhood.

### B. Bound Derivation

There are a few interesting cases for planted clique detection, which consider different growth rates for the background probability. First, consider the case where the background probability remains constant as the graph grows. In this scenario, the average degree of the graph grows linearly with $N$. In this case, the distribution $\mathcal{B}(N-k,p)$ will approach a normal distribution $\mathcal{N}((N-k)p,(N-k)p(1-p))$. We want to determine when the $k-1$ clique in the cue's neighborhood will be discovered with high probability, meaning that

$$(k-1)(1-p) > \sqrt{(k-1+\hat{d}_c)p(1-p)}. \tag{7}$$

Assuming $k = o(N)$, the $\hat{d}_c$ term will dominate. As $N$ grows, $(N-k)p+C\sqrt{N-k}$ for a constant $C$ will be a fixed number of standard deviations from the mean of $\hat{d}_c$, meaning that $\hat{d}_c$ will take on values greater than this with fixed probability. Thus, by considering a threshold $(N-k)p+C(N-k)^{0.5+\delta}$, $0.5 < \delta < 1$, we capture a polynomially increasing number of standard deviations, which will result in an exponential reduction in the probability of $\hat{d}_c$ crossing the threshold as $N$ increases. The asymptotic bound, therefore is

$$k > \sqrt{\frac{Np^2}{1-p}}.$$

The threshold value for the clique size still scales as the square root of the number of total vertices, but it can be a constant factor ($\sqrt{p}$) smaller than in the uncued case.

In practice, graphs typically do not increase their average degree linearly as the number of vertices increases. Studies have shown that the average degree often follows a sublinear polynomial [6]. Thus, it is also important to consider cases where the average degree $d_{\text{avg}}$ is $O(N^\delta)$, $0 < \delta < 1$. In this scenario, $p = O(N^{\delta-1})$, so the $(1-p)$ terms in (7) will approach 1. By a similar argument to the constant $p$ case, assuming the neighborhood size is a sublinearly increasing number of standard deviations above the mean, we asymptotically approach a detection threshold of

$$k > \sqrt{CN^{2\delta-1}} \tag{8}$$

for a constant $C$. In this case, it is possible that the detectable clique size can actually get *smaller* as the graph grows, since the one-hop neighborhood, although it grows slowly, is sparser. Using a variable $\delta$ helps demonstrate behavior for various growth patterns: If the density is maintained ($\delta \longrightarrow 1$), the minimum detectable clique size grows as the square root of the size of the graph, whereas if the average degree grows very slowly ($\delta \longrightarrow 0$), the size of the clique can decrease at a rate close to $1/\sqrt{N}$ and be detected by the cued method.

### C. Extension to Dense Subgraphs

Considering dense subgraphs rather than cliques, it may not be the case that the entire subgraph is in the one-hop neighborhood. One interesting question in this case is when multiple hops improve detectability. For the sake of simplicity, consider the expected value of the neighborhood size, $\mathbb{E}[\hat{d}_c] = kp_{in} + (N-k)p$, which, for large $N$, will be approximately $Np$. The number of additional background nodes added in the second hop is approximated by $(N-k-Np)(1-(1-p)^{Np})$. For small $p$ and large $N$, this is asymptotically quadratic in $p$ and $N$, behaving like $\Theta(N^2p^2)$, i.e., the average degree squared. For large $p_{in}$, most of the dense subgraph will be captured in the first hop, and the additional vertices will hurt performance. If, on the other hand, the subgraph edge probability is relatively small, then multiple hops will similarly expand the number of subgraph vertices available for the spectral algorithm to detect. The number of subgraph vertices gained from neighbors within the subgraph is $O(k^2p_{in}^2)$, and the number gained from external neighbors is $O(Nkp^2)$. The planted clique will be detectable in the two-hop neighborhood if either $k^2p_{in}^3$ or $Nkp^2p_{in}$ grows faster than $Np^{3/2}$.

## V. Experiments

In the previous section we theoretically analyzed the performance gain achieved by providing a cue vertex to the spectral method. In recognition of the importance of localized community detection approaches, there has been a proliferation of techniques that follow different perspectives of incorporating partial knowledge in the solution. We consider a few representative techniques from this literature, two local spectral algorithms (MOVCUT and Quadratic Programming), and two local random walk algorithms (Approximate Personalized PageRank and Threat propagation). We compare their empirical performance to the cue-based spectral method in both the clique and subgraph detection setting, for various degrees of problem difficulty. We first give a brief description of each algorithm and show in Section VI that the different notions of locality they utilize lead to different empirical performance for hard to detect cases.

### A. MOVCUT

The MOVCUT algorithm [7] extends the traditional spectral clustering formulation by adding a constraint that only considers solution vectors $\mathbf{x}$ that correlate well with the seed vector $\mathbf{s}$. Given a correlation parameter $k$, the local spectral optimization problem is written as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T L \mathbf{x} \\ \text{subject to} \quad & \mathbf{x}^T \mathbf{x}^T = 1, \\ & \mathbf{x}^T D^{1/2} \mathbf{1} = 0, \\ & (\mathbf{x}^T D^{1/2} \mathbf{s})^2 \geq k. \end{aligned}$$

The solution vector is expressed by:

$$\mathbf{x}^* = c(L - \gamma D)^+ D\mathbf{s},$$

where $c \in [0, \infty]$ is a normalization constant to make the solution $\mathbf{x}^*$ a unit normed vector, and $\gamma \in (-\infty, \lambda_2(G))$ ensures that $\mathbf{x}^*$ is found exactly on the boundary of the feasible region. [7] showed that sweeping through the locally biased solution $\mathbf{x}^*$ has analogous theoretical guarantees to the traditional spectral clustering solution. The MOVCUT algorithm combines both global and local aspects of graph structure. The $\mathbf{x}^*$ vector is still a solution to a global optimization problem, yet the restriction that it correlates to the seed by at least $k$ ensures that volume of the cut is no bigger than $k$ therefore localizing the output.

### B. Quadratic Programming

The quadratic programming algorithm is another local spectral algorithm that we consider. In contrast to MOVCUT, it uses the modularity matrix which emphasizes the fact that we would like to identify subgraphs with unexpected density (relative to some null distribution). In addition, the objective of this algorithm directly incorporates the knowledge of the seed in the solution vector. Formally, the quadratic programming algorithm optimizes the following:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T (\rho I - B) \mathbf{x} \\ \text{subject to} \quad & x_i \leq 0, i \neq s, \\ & x_s \leq -1, \end{aligned}$$

where $\rho$ is its 2-norm of the modularity matrix $B$.

### C. Approximate Personalized PageRank (APP)

Within the class of random walk partitioning algorithms, the personalized PageRank algorithm [8] has been used to rank the importance of vertices relative to an input seed vertex $\mathbf{s}$. The solution to the personalized PageRank problem is expressed as follows:

$$\mathbf{r} = \alpha \mathbf{s} + (1 - \alpha) D^{-1} A \mathbf{r},$$

where $\alpha$ is the teleportation probability to $\mathbf{s}$. This solution can be re-written in the form:

$$\mathbf{r} = (L + \frac{1 - \alpha}{\alpha} D)^{-1} D\mathbf{s}$$

to emphasize the connection between the spectral and random walk solutions with $\gamma = -\frac{1-\alpha}{\alpha}$.

Andersen et al. [9] developed an algorithm that approximates the personalized PageRank solution by iteratively distributing probabilities (vertex ranking scores) in a way that favors the region near the seed vertex. The PageRank solution $\mathbf{r}$ is expressed as an approximate vector $\tilde{\mathbf{r}}$ plus a residual vector $\mathbf{e}$: $\mathbf{r} = \tilde{\mathbf{r}} + \mathbf{e}$. The initial residual vector is the indicator vector for seed $s$. Given $s$, the algorithm moves an $\alpha$ fraction of the probability from $e_s$ to $\tilde{r}_s$. It then distributes the remaining $1 - \alpha$ probability, half to itself and half to its neighbors in magnitude proportional to their degree. The algorithm repeats until a large portion of the probability has been pushed back to the approximate solution vector $\tilde{\mathbf{r}}$. [9] showed that sweeping through their local approximation PageRank vector offers similar guarantees to known Cheeger inequality. Note that the APP algorithm is a true local algorithm in that it only uses local knowledge of the neighborhood around a vertex to update PageRank scores.

### D. Threat Propagation

Threat propagation algorithm [10] is similar to the class of personalized PageRank algorithms, but has the following distinguishing features. It views the graph partitioning problem as a $2^N$ multiple hypothesis test problem, where membership (to the cut set) or non-membership needs to be determined for all the vertices. It maximizes the Bayesian probability of detection by computing the harmonic solution to Laplace's equation, but treats this as a boundary value problem with seeds representing the boundary values and unknown values representing the interior. Also, instead of considering a constant diffusion probability $1 - \alpha$, it considers non-uniform diffusion probabilities inversely proportional to the average path lengths between the seed vertex and other vertices. This modification biases diffusion towards regions of the graph that are tightly connected to the seed vertex, therefore implicitly leading to localized, sparse solutions around the seed. The algorithm is proved to be optimum in the Neyman-Pearson sense of maximizing the probability of detection at a fixed false alarm probability [10].

## VI. Empirical Results

Results, in the form of an ROC curve for a single parameter setting, are given in Figure 1. This figure shows the average ROC curves of each methodology over 100 graphs with $|V| = 1000$ and $p_{ER} = 0.2$. The mean and standard deviations of the area under the ROC curves across all parameter settings are given in Table I. Best performing methodologies in a specific parameter setting are highlighted in bold. This table outlines the algorithmic performance for the 20 vertex planted clique and planted dense supgraph experiments. In these results, we observe good detection performance on the sparser background networks ($p = 0.1$ to $p = 0.2$). Generally, on these relatively simple problems, variation on detection performance as measured by mean AUC is low.

For the planted dense subgraph experiment, we observe that as networks grow more dense, the most effective methodology shifts from the quadratic programming based algorithm to Approximate PageRank. In the planted clique experiments, we observe the same two algorithms generally outperforming the rest.

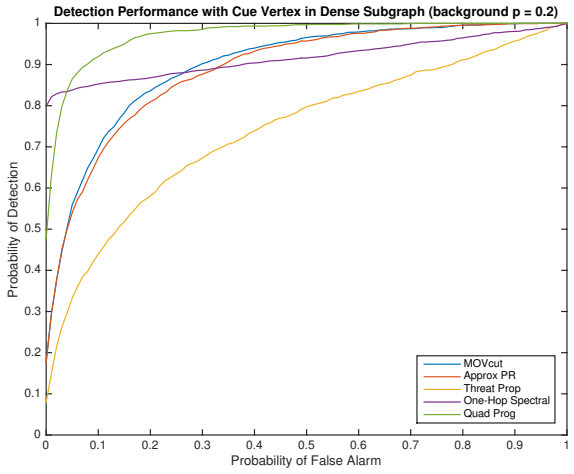


Fig. 1. An ROC curve showing detection performance of vertices in the neighborhood of a cue vertex embedded in a $p_{in} = 0.8$ subgraph. The background network generated is an Erdős-Rényi network with an edge probability of 0.2. Line colors represent the performance of different methodologies.

## VII. Summary

In this paper, we extend recent bounds based for planted clique detection to cases where one of the clique vertices is revealed. We show that this reduces to the problem of finding a smaller clique within a smaller random background, and that the same random matrix theory analysis holds after an initial filtering of the vertices. The resulting bounds show that a clique can be detected that grows more slowly than required in the uncued case by a factor of the edge probability, which implies that, when the average degree grows very slowly, smaller cliques can be detected as the total number of vertices increases. Considering 4 cued subgraph detection methods from the open literature, we show that a phase transition occurs for these methods as it also occurs for the simple method of applying a spectral detection method to the one-hop neighborhood of the graph.

From this point, a number of future directions are possible for this research. Understanding the limits of cued detection in graphs with community structure and arbitrary degree distributions is one important area. This will be complicated by the dependence on where the subgraph is placed (on high- or low-degree vertices, within a single background community or across several, etc.). Another interesting result would be to consider methods for proving cued detectability not relying on the same random matrix theoretic analysis as used here. It is possible that, for flow-based algorithms, other analytic techniques may be more appropriate. Even considering random matrix theory techniques, it would be ideal to compute a bound directly for the cued case, rather than using the uncued bound on a filtered subset.

## References


[1] G. A. Coppersmith and C. E. Priebe, "Vertex nomination via content and context," 2012, preprint: arXiv.org:1201.4118.

[2] B. A. Miller, M. S. Beard, P. J. Wolfe, and N. T. Bliss, "A spectral framework for anomalous subgraph detection," *IEEE Trans. Signal Process.*, vol. 63, no. 16, pp. 4191–4206, Aug 2015.

[3] E. Arias-Castro and N. Verzelen, "Community detection in random networks," 2013, preprint: arXiv.org:1302.7099.

[4] R. R. Nadakuditi, "On hard limits of eigen-analysis based planted clique detection," in *Proc. IEEE Statistical Signal Process. Workshop*, 2012, pp. 129–132.

[5] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, no. 3, 2006.

[6] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shinking diameters," *ACM Trans. Knowledge Discovery from Data*, vol. 1, no. 1, 2007.

[7] M. W. Mahoney, L. Orecchia, and N. K. Vishnoi, "A spectral algorithm for improving graph partitions," *CoRR*, vol. abs/0912.0681, 2009.

[8] G. Jeh and J. Widom, "Scaling personalized web search," in *Proceedings of the 12th International Conference on World Wide Web*, ser. WWW '03. New York, NY, USA: ACM, 2003, pp. 271–279.

[9] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using pagerank vectors," in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '06, 2006, pp. 475–486.

[10] S. T. Smith, E. K. Kao, K. D. Senne, G. Bernstein, and S. Philips, "Bayesian discovery of threat networks," *IEEE Trans. Signal Process.*, vol. 62, pp. 5324–5338, October 2014.

TABLE I
TABLE OF AUC MEANS (STANDARD DEVIATIONS) FOR 20 VERTEX PLANTED SUBGRAPH

| | Planted Dense Subgraph $p_{in} = 0.8$ | | | | | |
|---|---|---|---|---|---|---|
| | $p$ | | | | | |
| Method | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 |
| ApproxPR | 0.972 (0.021) | 0.863 (0.056) | 0.785 (0.074) | **0.733** (0.096) | **0.718** (0.114) | **0.692** (0.120) |
| MovCut | 0.997 (0.009) | 0.837 (0.086) | 0.689 (0.085) | 0.635 (0.097) | 0.563 (0.084) | 0.556 (0.071) |
| OneHop | 0.900 (0.058) | 0.887 (0.071) | 0.663 (0.151) | 0.549 (0.083) | 0.513 (0.067) | 0.511 (0.053) |
| QuadProg | **1.000** (0.0001) | **0.947**(0.046) | **0.811** (0.073) | 0.701 (0.077) | 0.615 (0.088) | 0.575 (0.095) |
| ThreatProp | 0.835 (0.182) | 0.722 (0.147) | 0.646 (0.121) | 0.618 (0.086) | 0.566 (0.091) | 0.538 (0.089) |
| | Planted Clique | | | | | |
| | $p$ | | | | | |
| Method | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 |
| ApproxPR | 0.999 (0.001) | 0.946 (0.024) | 0.861 (0.053) | 0.793 (0.084) | 0.728 (0.117) | 0.698 (0.107) |
| MovCut | **1** (0) | 0.992 (0.011) | 0.874 (0.077) | 0.750 (0.083) | 0.678 (0.078) | 0.617 (0.086) |
| OneHop | **1** (0) | **1** (0) | **1.000** (0.0003) | **0.901** (0.135) | 0.558 (0.128) | 0.519 (0.088) |
| QuadProg | **1** (0) | **1.000** (0.0005) | 0.980 (0.016) | 0.892 (0.053) | **0.797** (0.075) | **0.699** (0.077) |
| ThreatProp | 0.921 (0.123) | 0.883 (0.112) | 0.760 (0.136) | 0.705 (0.094) | 0.656 (0.084) | 0.607 (0.074) |